JetBox 8210 User Manual

Linux

www.korenix.com

0.0.2

Copyright Notice

Copyright© 2008 Korenix Technology Co., Ltd.
All rights reserved.
Reproduction without permission is prohibited.

Information provided in this manual is intended to be accurate and reliable. However, the original manufacturer assumes no responsibility for its use, or for any infringements upon the rights of third parties that may result from its use. The material in this document is for product information only and is subject to change without notice. While reasonable efforts have been made in the preparation of this document to assure its accuracy, Korenix assumes no liabilities resulting from errors or omissions in this document, or from the use of the information contained herein.
Korenix reserves the right to make changes in the product design without notice to its users.

Acknowledgments

Korenix is a registered trademark of Korenix Technology Co., Ltd.
All other trademarks or registered marks in the manual belong to their respective manufacturers.

Table of Content

Copyright Notice 2
Acknowledgments 2
Table of Content 3
Chapter 1 Overview 5
1-1 Software specification 5
1-2 Software service 6
1-3 Deploy a new Linux image on a CF card 8
Chapter 2 Getting Started 8
2-1 Setting up the JetBox directly (VGA console) 9
2-2 Connecting the JetBox with a PC (RS-232 or telnet via a network) 9
2-3 Booting up 10
2-4 Using it 10
2-5 Configuring network 11
Chapter 3 Using the JetBox Linux 12
3-1 System administration guide 12
3-1-1 etc/inittab 12
3-1-2 Scripts in /etc/init.d/ 13
3-1-3 etc/X11/XF86Config 13
3-1-4 Files in /usr/share/fvwm 13
3-2 Base software package 13
3-2-1 busybox 13
3-2-2 GNU bash 14
3-2-3 e2fsprogs 15
3-2-4 dosfstools 15
3-2-5 Setserial 16
3-3 Libraries 17
3-3-1 libxml2 17
3-3-2 libusb 17
3-3-3 DirectFB 17
3-3-4 SDL 18
3-3-5 freetype 18
3-4 Network related software packages 18

3-4-1 bridge-utils 18
3-4-2 iptables 20
3-4-3 net-snmp 20
3-4-4 ntp 21
3-4-5 dropbear 21
3-4-6 openvpn 22
3-4-7 openswan 23
3-4-8 pppd 23
3-4-9 rp-pppoe 24
3-4-10 pptp-linux 24
3-4-11 proftpd 24
3-4-12 samba 25
3-4-13 tcpdump 25
3-4-14 php 26
3-4-15 apache 27
3-4-16 dillo 27
3-4-17 rxvt 28
3-4-18 mplayer 28
3-4-19 fvwm 29
3-4-20 lbreakout 30
3-4-21 ltris 30
3-4-22 mp3play 31
3-4-23 rdesktop 31
Chapter 4 SDK (Software Development Kits) 31
4-1 Installing SDK 31
4-2 Writing your application 32
4-3 Sample programs 32
4-3-1 DIO access 33
4-3-2 RS-232/422/485 control 36
Chapter 5 Appendix 39
5-1 Notice, Chart & Example Index 39
5-2 Customer Service 40

Chapter 1 Overview

The advantage of adopting Korenix JetBox series is ready-to-use. Korenix is devoted to improve the usability of embedded computer in industrial domain. Besides operating system (Linux/ WinCE), Korenix provides device drivers, protocol stacks, system utilities, supporting services and daemons in one CompactFlash card to make system integration simple. Further, Korenix provides application development toolkits for users to build up their own applications easily.

JetBox 8210 is a high performance, compact and rugged embedded computer. All-in-one device with small volume, fanless design and a capability to withstand a wide range of temperatures is suitable for industrial severe environment. It is equipped with Intel Xscale PXA270 RISC processor and 128MB SDRAM and supports Linux and WinCE5.0 to meet requirements of industrial PC applications. For better expansibility, it carries 4 USB ports, 2 RS-232 ports and 2 RS-232/422/485 ports for versatile peripheral and interfaces and one CompactFlash slot for system integration. It also supports VGA and audio to give users much flexibility in industrial applications. In addition, it is equipped with 2 RJ-45 ports and supports daemons and web server services to accommodate to the network communication environment today.

With complete SW solution and excellent HW design, JetBox series is the best choice of embedded computer.

1-1 Software specification

Embedded Linux 2.6.18	
Protocol Stack	TCP, UDP, IPv4, SNMP, ICMP, IGMP, ARP, HTTP, CHAP, PAP, SSH1.0/2.0, SSL, DHCP, NTP, NFS, SMTP, Telnet, FTP, PPP, PPPoE
System Utilities	Bash, busybox, tinylogin, telnet, ftp
Supporting Services and Daemons	**telnetd:** Telnet server daemon **ftpd:** FTP server daemon **sshd:** Secure shell server

	Apache: Web server daemon, supports PHP and XML **MySQL:** Database server and client **OpenVPN:** Virtual private network service manager **iptables:** Firewall service manager **pppd:** Dial in/out over serial port daemon **snmpd:** snmpd agent daemon **inetd:** TCP server manager program
Application Development Environment	Korenix Linux API library Linux tool chain: Gcc, Glibc
Device Drivers	CF card, USB, Watchdog timer, UART, VGA, Audio

Chart 1: The JetBox 8210 Linux SW specification

1-2 Software service

Item	Notes	JetBox 8210
Boot Loader		JetBox bootloader with CF booting support
Kernel		Linux 2.6.18
File System		**Ext3**
Programming API		(1) GPIO access (2) RS-232/422/485 configuration (3) Watchdog API
Base SW package		
Shell	OS shell command	GNU Bash3.2
Busybox	Linux normal command utility	1.8.2
e2fsprogs	Ext2/Ext3 file system utilities	1.39
dosfstools	DOS file system utility	2.11
setserial	RS-232 serial port setting tool	2.17
Libraries		
libxml2	XML library	2.6.29
libusb	USB support library	0.1.12
DirectFB	Frame Buffer access library	1.0.1

Item	Notes	JetBox 8210
SDL	SDL multimedia library	1.2.12
freetype	TrueType font support library	2.2.1
Network related SW package		
bridge-utils	Ethernet bridge utility	1.0.6
iptables	NAT setting tool	1.3.8
net-snmp	SNMP support package	5.1.2
ntp	NTP utility	4.2.0
dropbear	SSH support package	0.50
openvpn	VPN tool	2.0.9
openswan	Ipsec for Linux	2.4.9
pppd	PPP protocol for Linux	2.4.4
rp-pppoe	PPPOE support package	3.8
pptp-linux	PPTP protocol for Linux	1.7.0
proftpd	FTP daemon	1.3.0
samba	SMB (Windows network) support package	3.0.26a
tcpdump	A tool for network monitoring and data acquistion	3.9.5
php	Web scripting language	5.2.3
apache	Web server	1.3.39
mySQL	database	5.0.51
Graphic and Multimedia		
XFree	X window system	4.5.0
dillo	A light weight Web Browser	0.8.6
rxvt	A nice small colour vt102 X terminal emulator	2.7.5
mplayer	A movie player	1.0rc1
fvwm	A window manager for X window	2.5.18
lbreakout	A Linux breakout game	--

Item	Notes	JetBox 8210
ltris	A Linux tetris game	--
Linux tool chain		
Gcc	C/C++ PC Cross Compiler	
Glibc	POSIX standard C library	2.3.x

Chart 2: The JetBox 8210 Linux SW service list

1-3 Deploy a new Linux image on a CF card

The file "**jetbox_rootfs.taz**" is the root filesystem archive of the JetBox. You can use this file to create a new booting CF card by following steps:

1. Insert a empty CF card into a Linux PC
2. Create a primary partition on the CF card. The boot partition should be the first partition and partition type ID is **0x83 (Linux)**. The partition size should be large enough to put the root filesystem into it.
3. Format the partition created by step2 as **ext2** or **ext3** filesystem.
4. Mount the partition created by step2 into your Linux PC, un-tar jetbox_rootfs.tgz into it.
5. Unmount the CF card, put the CF card into the JetBox and boot the JetBox. The New CF image is applied in the JetBox.

Chapter 2 Getting Started

The JetBox has 3 ways to set up the configuration at the first time, by VGA console, or connecting with a PC by RS-232, or by telnet via a network.

To log in, type the Login name and password as requested.
The default values as following.
Login: root
Password: none

You can also set your root password by the command "**passwd.**"

2-1 Setting up the JetBox directly (VGA console)

Connect the VGA output to a VGA monitor and connect USB ports to an USB keyboard and an USB mouse to control the JetBox directly.

2-2 Connecting the JetBox with a PC (RS-232 or telnet via a network)

By RS-232

Connect one RS-232 port of the JetBox to your PC by a RS-232 null modem cable.
The default setting of the JetBox Linux for RS-232 ports:

Item	Parameter
Baud rate	115200bps
Parity	None
Data bits	8
Stop bits	1
Flow Control	None

Chart 3: The default RS-232 setting of the JetBox 8210 LInux

In the factory default setting, the **getty** command is running for all four RS-232 ports to wait for user login. You can disable RS-232 login by editing **/etc/inittab** .

By telnet

Firstly, you should modify the IP address and subnetmask of your PC to make your PC be in the same subnet as the JetBox 8210.
To connect the JetBox 8210 to your PC directly, use a cross-over Ethernet cable.
To connect the JetBox 8210 to your local LAN via a hub or a switch, use a straight-through Ethernet cable.

The default IP addresses and subnetmasks are shown as follows:

	Default IP address	Subnetmask

LAN1	eth0	192.168.10.1	255.255.255.0
LAN2	eth1	Assigned by DHCP server	

Chart 4: The default IP address setting of the JetBox 8210 Linux

2-3 Booting up

When you power on the JetBox, a splash screen appears. It takes around 30 to 60 seconds to boot up the JetBox Linux. After the booting process is completed, an X window system with FVWM (F virtual window manager) will show on the screen and then you can start to use the JetBox.

⚠**Notice 1:** If the image of the CF card is incorrect or there is no CF card in the JetBox, the splash screen shows a warning message “check CF card??”

If you connect a USB mouse and a USB keyboard to the JetBox, you can use the opened RXVT terminal (Linux shell) or open a new one by clicking the “Terminal” icon.

You can also switch to Linux virtual console by the hot key “Ctrl-Alt-F1” and switch back to X window system by the hot key “Ctrl-Alt-F3.”

2-4 Using it

The JetBox Linux environment is similar with the ordinary Linux distributions. You will be familiar with the JetBox if you are used to the ordinary Linux distributions.

All the command style, philosophy and configuration method are the same as the Linux distribution. Besides X window system, you can also use the shell in the same way as ordinary Linux distributions.

In the factory default setting, the JetBox Linux will boot into X window system with FVWM as the window manager. There are some demo programs on the right panel of the screen.

- Mplayer: a video player demo
- Terminal: To open a RXVT terminal for bash shell command
- Xterm: To open an Xterm window for bash shell command

- Tetris: A Tetris game
- Breakout: A breakout game
- Breakout 2: A breakout game
- Dillo: A web browser
- Xcalc: A calculator

2-5 Configuring network

You can change your network configuration by editing the file /etc/network/interfaces.

Following is a sample network configuration. This configuration is to set eth0 to a static IP 192.168.10.1 and get the IP of eth1 from DHCP server when the JetBox boots up. If you make any changes in this file, you should reboot the JetBox to make the configuration active.

```
# Configure Loopback
auto eth0 eht1 lo

iface lo inet loopback

#
iface eth0 inet static
     address 192.168.10.1
     network 192.168.10.0
     netmask 255.255.255.0
     broadcast 192.168.10.255

iface eht1 inet dhcp
```

You can also use following commands to configure the network setting for the JetBox.

Ifconfig (interface configure)

To identify the network interface to TCP/IP and to assign the IP address, subnetmask, and broadcast address to the interface.

udhcpc
To get an IP address from DHCP server

Chapter 3 Using the JetBox Linux

The Jetbox Linux is similar with ordinary Linux distributions. An experienced Linux user gets used to the JetBox quickly.
For a Linux novice, a good tutorial book for reference will be helpful and http://www.tldp.org/ is a great website for reference. There are a lot of Linux related information.

In this chapter, it introduces the software services in the JetBox and this helps you have an overall picture about the JetBox Linux.

3-1 System administration guide

This section provides you the information to customize the JetBox.

3-1-1 /etc/inittab

It is the configuration file that controls the startup procedure of the JetBox.

::once: /etc/init.d/xfree
To start up the X window system
You can comment out the line to disable X window system.

::respawn:/sbin/getty 38400 tty1
To control the virtual consoles with login shell created.

ttyS0::respawn:/sbin/getty –L ttyS0 115200 vt100
To enable or disable the login from a certain serial port (RS-232 mode)

3-1-2 Scripts in /etc/init.d/

Scripts in /etc/init.d/ control the startup of certain daemons. You can add your own startup scripts into this directory. Only a script starting with the character “S” will be used. And the sequence of startups depends on the number after the “S” character. For example:
There are two startup scripts for ntp daemon and proftpd daemon.

/etc/init.d/S49ntp

/etc/init.d/S50proftpd

S50 runs after S49. And if you don’t want ntp to be started while the JetBox boots up, you can just rename /etc/init.d/S49ntp to /etc/init.d/ntp. Or if you want to modify the startup sequence of “ntp”, you just modify the number after the “S” character, such as /etc/init.d/S51ntp.

3-1-3 /etc/X11/XF86Config

This file is the configuration file for X window system. It controls the operation of X window system.

3-1-4 Files in /usr/share/fvwm

Those files control the operation of the window manger fvwm in X window system. You can edit the file “**/usr/share/fvwm/system.fvwm2rc**” to control the operation of fvwm and the startup application of X window system.

3-2 Base software package

3-2-1 busybox

BusyBox combines tiny versions of many common UNIX utilities into a single small executable. It provides replacements for most of the utilities you usually find in GNU fileutils, shellutils, etc. The utilities in busybox generally have fewer options than their full-featured GNU cousins; however, the options that are included provide the expected functionality and behave very much like their GNU counterparts. Busybox provides a fairly complete environment for any small or embedded system.

Busybox has been written with size-optimization and limited resources in mind. It is also extremely modular so you can easily include or exclude commands (or features) at compile time. This makes it easy to customize your embedded systems. To create a working system, just add some device nodes in /dev, a few configuration files in /etc, and a Linux kernel.

The JetBox Linux use busybox for most of the base Linux utility. The following is a list of busybox command in the JetBox Linux.

[, [[, addgroup, adduser, ash, basename, bunzip2, busybox, bzcat, cat, chgrp, chmod, chown, chroot, chvt, clear, cmp, cp, cut, date, dc, dd, deallocvt, delgroup, deluser, df, diff, dirname, dmesg, dos2unix, du, dumpleases, e2fsck, echo, egrep, eject, env, expr, false, fbset, fdformat, fdisk, fgrep, find, free, fsck.ext2, fsck.ext3, ftpget, ftpput, getopt, getty, grep, gunzip, gzip, halt, hdparm, head, hexdump, hostid, hostname, hwclock, id, ifconfig, ifdown, ifup, init, install, ip, kill, killall, klogd, linuxrc, ln, logger, login, logname, ls, md5sum, mkdir, mke2fs, mkfs.ext2, mkfs.ext3, mknod, mktemp, more, mount, mt, mv, netstat, nslookup, openvt, passwd, pidof, ping, pivot_root, poweroff, ps, pwd, rdate, readlink, reboot, reset, rm, rmdir, route, run-parts, sed, sh, sha1sum, sleep, sort, start-stop-daemon, strings, stty, su, sulogin, swapoff, swapon, sync, syslogd, tail, tar, tee, telnet, telnetd, test, tftp, time, touch, true, tty, udhcpc, udhcpd, umount, uname, uniq, unix2dos, unzip, uptime, usleep, uudecode, uuencode, vi, vlock, wc, wget, which, whoami, xargs, yes, zcat

You can get more information about busybox at http://www.busybox.net/.

3-2-2 GNU bash

Bash is the shell, or command language interpreter, that will appear in the GNU operating system. Bash is a sh-compatible shell that incorporates useful features from the Korn shell (ksh) and C shell (csh). It is intended to conform to the IEEE POSIX P1003.2/ISO 9945.2 Shell and Tools standard. It offers functional improvements over sh for both programming and interactive use. In addition, most sh scripts can be run by bash without modification.

The improvements offered by bash include:

- Command line editing
- Unlimited size command history

- Job control
- Shell functions and aliases
- Indexed arrays of unlimited size
- Integer arithmetic in any base from two to sixty-four

The JetBox Linux uses bash as the default shell. You can get more information about bash at http://www.gnu.org/software/bash/.

3-2-3 e2fsprogs

The e2fsprogs package contains a number of utilities for creating, checking, modifying, and correcting any inconsistencies in second extended (ext2) or ext3 filesystems. E2fsprogs contains e2fsck (used to repair filesystem inconsistencies after an unclean shutdown), mke2fs (used to initialize a partition to contain an empty ext2/ext3 filesystem), debugfs (used to examine the internal structure of a filesystem, to manually repair a corrupted filesystem or to create test cases for e2fsck), tune2fs (used to modify filesystem parameters), resize2fs to grow and shrink unmounted ext2/ext3 filesystems, and most of the other core ext2fs filesystem utilities.

The e2fsprogs package includes libuuid library, libblkid library and fsck tool.
The filesystem utilities for the EXT2/EXT3 filesystem include e2fsck, mke2fs, dumpe2fs, fsck…etc. And you can get more information at http://e2fsprogs.sourceforge.net/.

3-2-4 dosfstools

The package includes 2 utilities, mkdosfs and dosfsck.

Mkdosfs is used to create an MS-DOS file system under Linux on a device (usually a disk partition). Device is the special file corresponding to the device (e.g /dev/hdXX). Block-count is the number of blocks on the device. If omitted, mkdosfs automatically determines the file system size.

Dosfsck verifies the consistency of MS-DOS file systems and optionally tries to repair them. The following file system problems can be corrected (in this order):

- FAT contains invalid cluster numbers. Cluster is changed to EOF.

- File's cluster chain contains a loop. The loop is broken.
- Bad clusters (read errors). The clusters are marked bad and they are removed from files owning them. This check is optional.
- Directories with a large number of bad entries (probably corrupt). The directory can be dropped.
- Files “.” and “..” are non-directories. They can be dropped or renamed.
- Directories “.” and ”..” in root directory. They are dropped.
- Bad file names. They can be renamed.
- Duplicate directory entries. They can be dropped or renamed.
- Directories with non-zero size field. Size is set to zero.
- Directory “.” does not point to parent directory. The start pointer is adjusted.
- Directory “..” does not point to parent of parent directory. The start pointer is adjusted.
- Start cluster number of a file is invalid. The file is truncated.
- File contains bad or free clusters. The file is truncated.
- File's cluster chain is longer than indicated by the size fields. The file is truncated.
- Two or more files share the same cluster(s). All but one of the files are truncated. If the file being truncated is a directory file that has already been read, the file system check is restarted after truncation.
- File's cluster chain is shorter than indicated by the size fields. The file is truncated.
- Clusters are marked as used but are not owned by a file. They are marked as free.

Additionally, the following problems are detected, but not repaired:

- Invalid parameters in boot sector.
- Absence of “.” and “..” entries in non-root directories.

3-2-5 Setserial

Setserial is a program designed to set and/or report the configuration information associated with a serial port. This information includes what I/O port and IRQ a particular serial port is using, and whether or not the break key should be interpreted as the Secure Attention Key, and so on.

You can get more information at http://setserial.sourceforge.net/.

3-3 Libraries

3-3-1 libxml2

Libxml2 is the XML C parser and toolkit developed for the Gnome project (but usable outside of the Gnome platform), it is free software available under the MIT License. XML itself is a metalanguage to design markup languages, i.e. text language where semantic and structure are added to the content using extra "markup" information enclosed between angle brackets. HTML is the most well-known markup language. Though the library is written in C a variety of language bindings make it available in other environments.

You can get more information at http://xmlsoft.org/.

3-3-2 libusb

Libusb is a set of USB API for users' applications to directly control a USB device. The library supports users to write some USB device utilities.

You can get more information at http://libusb.wiki.sourceforge.net/.

3-3-3 DirectFB

DirectFB is a thin library that provides hardware graphics acceleration, input device handling and abstraction, integrated windowing system with support for translucent windows and multiple display layers, not only on top of the Linux Framebuffer Device. It is a complete hardware abstraction layer with software fallbacks for every graphics operation that is not supported by the underlying hardware. DirectFB adds graphical power to embedded systems and sets a new standard for graphics under Linux.

You can get more information at http://www.directfb.org/.

3-3-4 SDL

Simple DirectMedia layer is a cross-platform multimedia library designed to provide low level access to audio, keyboard, mouse, joystick, 3D hardware via OpenGL, and 2D video framebuffer.

SDL is written in C, but works with C++ natively, and has bindings to several other languages, including Ada, C#, Eiffel, Erlang, Euphoria, Guile, Haskell, Java, Lisp, Lua, ML, Objective C, Pascal, Perl, PHP, Pike, Pliant, Python, Ruby, and Smalltalk.

You can get more information at http://www.libsdl.org/.

3-3-5 freetype

Freetype is a software font engine that is designed to be small, efficient, highly customizable, and portable while capable of producing high-quality output (glyph images). It can be used in graphics libraries, display servers, font conversion tools, text image generation tools, and many other products as well.

Note that FreeType is a font service and doesn't provide APIs to perform higher-level features like text layout or graphics processing (e.g. colored text rendering, hollowing…etc.) However, it greatly simplifies these tasks by providing a simple easy to use, and uniform interface to access the content of font files.

You can get more information at http://www.freetype.org/.

3-4 Network related software packages

3-4-1 bridge-utils

A bridge is a way to connect two Ethernet segments together in a protocol independent way. Packets are forwarded based on Ethernet address, rather than IP address (like a router). Since forwarding is done at Layer 2, all protocols can go transparently through a bridge.

The Linux bridge code implements a subset of the ANSI/IEEE 802.1d standard. The original Linux bridging was first done in Linux 2.2, and then rewritten by Lennert Buytenhek. The code for bridging has been integrated into 2.4 and 2.6 kernel series.

The brctl command of bridge utility

```
# brctl
# commands:
    addbr           <bridge>              add bridge
    delbr           <bridge>              delete bridge
    addif           <bridge> <device>     add interface to bridge
    delif           <bridge> <device>     delete interface from bridge
    setageing       <bridge> <time>       set ageing time
    setbridgeprio   <bridge> <prio>       set bridge priority
    setfd           <bridge> <time>       set bridge forward delay
    sethello        <bridge> <time>       set hello time
    setmaxage       <bridge> <time>       set max message age
    setpathcost     <bridge> <port> <cost> set path cost
    setportprio     <bridge> <port> <prio> set port priority
    show                                  show a list of bridges
    showmacs        <bridge>              show a list of mac addrs
    showstp         <bridge>              show bridge stp info
    stp             <bridge> <state>      turn stp on/off
```

The following is an example of how to setup a bridge

```
# ifconfig eth0 0.0.0.0
# ifconfig eth1 0.0.0.0
# brctl addbr mybridge
# brctl addif mybridge eth0
# brctl addif mybridge eth1
# ifconfig mybridge up
```

You can get more information at http://www.linuxfoundation.org/en/Net:Bridge.

3-4-2 iptables

Iptables is the userspace command line program used to configure the Linux 2.4.x and 2.6.x IPv4 packet filtering ruleset. It is targeted towards system administrators.

Since Network Address Translation is also configured from the packet filter ruleset, iptables is used for this, too.

The iptables package also includes ip6tables. Ip6tables is used for configuring the IPv6 packet filter.

You can get more information at http://www.netfilter.org/.

3-4-3 net-snmp

Simple **N**etwork **M**anagement **P**rotocol (**SNMP**) is a widely used protocol for monitoring the health and welfare of network equipment (eg. routers), computer equipment and even devices like UPSs. Net-SNMP is a suite of applications used to implement SNMP v1, SNMP v2c and SNMP v3 using both IPv4 and IPv6. The suite includes:

- Command-line applications to:
  - retrieve information from an SNMP-capable device, either using single requests (snmpget, snmpgetnext), or multiple requests (snmpwalk, snmptable, snmpdelta).
  - manipulate configuration information on an SNMP-capable device (snmpset).
  - retrieve a fixed collection of information from an SNMP-capable device (snmpdf, snmpnetstat, snmpstatus).
  - convert between numerical and textual forms of MIB OIDs, and display MIB content and structure (snmptranslate).
- A graphical MIB browser (tkmib), using Tk/perl.
- A daemon application for receiving SNMP notifications (snmptrapd). Selected notifications can be logged (to syslog, the NT Event Log, or a plain text file),

forwarded to another SNMP management system, or passed to an external application.

- An extensible agent for responding to SNMP queries for management information (snmpd). This includes built-in support for a wide range of MIB information modules, and can be extended using dynamically loaded modules, external scripts and commands, and both the SNMP multiplexing (SMUX) and Agent Extensibility (AgentX) protocols.
- A library for developing new SNMP applications, with both C and perl APIs.

Net-SNMP is available for many Unix and Unix-like operating systems and also for Microsoft Windows.

Note: Functionality can vary depending on the operating system. Please see the README files for information specific to the JetBox platform.

You can get more information at http://net-snmp.sourceforge.net/.

The configuration file of net-snmp is located at **/etc/snmp**.

Example 1: Refer to the file—configuration example_snmpd.

3-4-4 ntp

NTP is a protocol designed to synchronize the clocks of computers over a network. NTP version 3 is an internet draft standard, formalized in RFC 1305. NTP version 4 is a significant revision of the NTP standard, and is the current development version, but has not been formalized in an RFC. Simple NTP (SNTP) version 4 is described in RFC 2030.

You can get more information at http://www.ntp.org/.

3-4-5 dropbear

Dropbear is a relatively small SSH 2 server and client. It runs on a variety of POSIX-based platforms. Dropbear is open source software, distributed under a MIT-style license. Dropbear is particularly useful for "embedded"-type Linux (or other

Unix) systems, such as wireless routers.

Features

- A small memory footprint suitable for memory-constrained environments - Dropbear can compile to a 110kB statically linked binary with uClibc on x86 (only minimal options selected)
- Dropbear server implements X11 forwarding, and authentication-agent forwarding for OpenSSH clients
- Can run from inetd or standalone
- Compatible with OpenSSH ~/.ssh/authorized_keys public key authentication
- The server, client, keygen, and key converter can be compiled into a single binary (à la busybox)
- Features can easily be disabled when compiling to save space
- TCP forwarding support

You can get more information at http://matt.ucc.asn.au/dropbear/dropbear.html.

3-4-6 openvpn

OpenVPN is a full-featured open source SSL VPN solution that accommodates a wide range of configurations, including remote access, site-to-site VPNs, Wi-Fi security, and enterprise-scale remote access solutions with load balancing, failover, and fine-grained access-controls.

OpenVPN implements OSI layer 2 or 3 secure network extension using the SSL/TLS protocol, supports flexible client authentication methods based on certificates, smart cards, and/or 2-factor authentication, and allows user or group-specific access control policies using firewall rules applied to the VPN virtual interface. OpenVPN is not a web application proxy and does not operate through a web browser.

You can get more information at http://openvpn.net/.

Openvpn's configuration files are at the directory **/etc/openvpn/.**

Example 2: Refer to the file—configuration example_openvpn.

3-4-7 openswan

Openswan is an IPsec implementation for Linux and BSD. It has support for most of the extensions (RFC + IETF drafts) related to IPsec, including X.509 Digital Certificates, NAT Traversal, and many others.

You can get more information at http://www.openswan.org/.

The configuration file of openswan is located at the directory **/etc/ipsec.d/** and **/etc/ipsec.conf**.

Example 3: Refer to the file—configuration example_openswan(ipsec).

3-4-8 pppd

The Point-to-Point Protocol (PPP) provides a standard way to establish a network connection over a serial link. At present, this package supports IP and the protocols layered above IP, such as TCP and UDP. The Linux and Solaris ports of this package have optional support for IPV6; the Linux port of this package also has support for IPX.

This software consists of two parts:

- Kernel code, which establishes a network interface and passes packets between the serial port, the kernel networking code and the PPP daemon (pppd). This code is implemented using STREAMS modules on Solaris, and as a line discipline under Linux.
- The PPP daemon (pppd), which negotiates with the peer to establish the link and sets up the ppp network interface. Pppd includes support for authentication, so you can control which other systems may make a PPP connection and what IP addresses they may use.

You can get more information at ftp://ftp.samba.org/pub/ppp/.

The authenticated key file and configuration files of pppd are located at the directory **/etc/ppp**.

3-4-9 rp-pppoe

PPPoE (Point-to-Point Protocol over Ethernet) is a protocol used by many ADSL Internet Service Providers. Roaring Penguin has a free PPPoE client for Linux and Solaris systems to connect to PPPoE service providers.

Dubbed RP-PPPoE, this open-source product is ideal for Linux users with a DSL "modem" whose Internet service provider uses PPPoE. Before you use this software, check whether or not you really need it. If your ISP uses PPPoE, but has given you a router, you may not need a PPPoE client on your Linux box. DHCP may work fine.

You can get more information at http://www.roaringpenguin.com/products/pppoe.

The configuration files of rp-pppoe are at the directory **/etc/ppp**.

Example 4: Refer to the file—configuration example_pppoe.

3-4-10 pptp-linux

PPTP Client is a Linux, FreeBSD, NetBSD and OpenBSD client for the proprietary Microsoft Point-to-Point Tunneling Protocol. PPTP allows connection to a PPTP based Virtual Private Network (VPN) as used by employers and some cable and ADSL internet service providers.

You can get more information at http://pptpclient.sourceforge.net/.

3-4-11 proftpd

ProFTPD is a highly configurable FTP daemon for Unix and Unix-like operating systems. See the README.ports file for more details about the platforms on which ProFTPD in known or thought to build and run.
ProFTPD grew from a desire for a secure and configurable FTP server. It was inspired by a significant admiration of the Apache web server. Unlike most other Unix ftp servers, it has not been derived from the old BSD ftpd code base, but is a completely new design and implementation.
ProFTPD's extensive configurability provides system administrators great flexibility in user authentication and access controls, including virtual ftp users and easy chroot()

ftp sessions for individual users.
ProFTPD is popular with many service providers for delivering update access to user web pages, without resorting to Unix shell accounts.

You can get more information at http://www.proftpd.org/.

The configuration file of proftpd is **/etc/proftpd.conf**.

Example 5: Refer to the file—configuration example_proftpd.

3-4-12 samba

Samba consists of two key programs. The two key programs are smbd and nmbd. Their job is to implement the four basic modern-day CIFS services, which are:

- File & print services
- Authentication and Authorization
- Name resolution
- Service announcement (browsing)

File and print services are, of course, the cornerstone of the CIFS suite. These are provided by smbd, the SMB Daemon. Smbd also handles "share mode" and "user mode" authentication and authorization. That is, you can protect shared file and print services by requiring passwords. In share mode, the simplest and least recommended scheme, a password can be assigned to a shared directory or printer (simply called a "share"). This single password is then given to everyone who is allowed to use the share. With user mode authentication, each user has their own username and password and the System Administrator can grant or deny access on an individual basis.

You can get more information at http://www.samba.org/.

The configuration files of samba are at the directory **/etc/samba**.

Example 6: Refer to the file—configuration example_samba.

3-4-13 tcpdump

Tcpdump is a tool for network monitoring and data acquisition. This software was originally developed by the Network Research Group at the Lawrence Berkeley

National Laboratory

You can get more information at http://www.tcpdump.org/.

3-4-14 php

PHP (recursive acronym for "PHP: Hypertext Preprocessor") is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML.

Following is an example:

```
Example#1 An introductory example
<html>
    <head>
        <title>Example</title>
    </head>
    <body>
        <?php
        echo "Hi, I'm a PHP script!";
        ?>
    </body>
</html>
```

Notice how this is different from a script written in other languages like Perl or C -- instead of writing a program with lots of commands to output HTML, you write an HTML script with some embedded code to do something (in this case, output some text). The PHP code is enclosed in special start and end tags that allow you to jump into and out of "PHP mode".

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server. If you were to have a script similar to the above on your server, the client would receive the results of running that script, with no way of determining what the underlying code may be. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer. Don't be afraid reading the long list of PHP's features. You can jump in, in a short time, and start writing simple scripts in a few hours.

Although PHP's development is focused on server-side scripting, you can do much more with it. Read on, and see more in the What can PHP do? section, or go right to the introductory tutorial if you are only interested in web programming.

You can get more information at http://www.php.net/.

Example 7: Refer to the file—configuration example_php.

3-4-15 apache

Apache is an HTTP server designed as a plug-in replacement for the NCSA server version 1.3 (or 1.4). It fixes numerous bugs in the NCSA server and includes many frequently requested new features, and has an API which allows it to be extended to meet users' needs more easily.

You can get more information at http://httpd.apache.org/.

The configuration file is at the directory **/etc/httpd.conf.**

Example 8: Refer to the file—configuration example_httpd.

3-4-16 dillo

- Dillo is a web browser project completely written in C.
- Dillo is small: source is less than 420 KB, and the binary is around 350 KB!
- Dillo aims to be a multi-platform browser alternative that's small, stable, developer-friendly, usable, fast, and extensible.
- Dillo is mainly based on GTK+ (GNOME is NOT required!)
- Dillo is a free-SW project in the terms of the GNU general public license.
- Dillo's bug meter is a tool to help towards standards compliance.
- Current code uses an improved html-parser (formerly based on gzilla's), and almost everything else was rewritten from scratch!.

- Dillo is very fast!

You can get more information at http://www.dillo.org/.

3-4-17 rxvt

Rxvt is a terminal emulator for the X Window System, originally written by Rob Nation and later extensively modified by Mark Olesen, who took over maintenance for several years. It is intended to be a slimmed-down replacement for xterm, omitting some of its little-used features, like Tektronix 4014 emulation and toolkit-style configurability. The latter refers to the Xt resource mechanism, e.g., for binding keys. Rxvt is an extended version of the older xvt terminal emulator by John Bovey of the University of Kent. The name originally stood for "Robert's xvt", but was later re-dubbed "our xvt" (pronounced like the letters r-x-v-t).

Aside from features such as those controlled by resource files, rxvt's terminal emulation differs from xterm in two important ways:

It emulates a VT102, rather than a VT220. That means that it handles 8-bit data differently, does not implement the C1 controls that xterm does. xterm does implement a switch "-k8" to suppress that functionality; rxvt does not provide an option to emulate a VT220.

The strings sent for function keys are different. xterm sends strings that are encoded using the same rules as the ANSI/ISO escape sequences. Rxvt's do not, though they provide comparable flexibility in this area.

You can get more information at http://www.rxvt.org/.

3-4-18 mplayer

MPlayer is a free and open source media player distributed under the GNU General Public License. The program is available for all major operating systems, including Linux and other Unix-like systems, Microsoft Windows and Mac OS X. Versions for OS/2, AmigaOS and MorphOS are also available. The Windows versions work, with some minor problems, also in DOS using HX DOS Extender. A port for DOS using

DJGPP is also available.

MPlayer supports a wide variety of media formats. In addition to its wide range of supported formats MPlayer can also save all streamed content to a file.

A companion program, MEncoder, can take an input stream or file and transcode it into several different output formats, optionally applying various transforms along the way.

MPlayer is a command line application which has different optional GUIs for each of its supported operating systems. Commonly used GUIs are gmplayer (the default GUI for GNU/Linux and other Unix-like systems, and Microsoft Windows), MPlayer OS X (for Mac OS X), MPUI (for Windows) and WinMPLauncher (also for Windows). Several other GUI frontends are also available for each platform.

You can get more information at http://www.mplayerhq.hu/.

3-4-19 fvwm

The F Virtual Window Manager is a virtual window manager for the X Window system. Originally a twm derivative, FVWM has evolved into a powerful and highly configurable environment for UNIX systems.

This is a partial list based on the documentation distributed with FVWM. Many of these features can be disabled at runtime or compile time, or dynamically for specific windows or loaded and unloaded as modules, or many other possibilities. These are not rigid features, FVWM does not dictate how the user's desktop should work or look like but provides the mechanisms to configure the desktop to work, look and behave the way the user wants it to.

- Supports any number of virtual desktops, each divided into multiple pages.
- Full EWMH, ICCCM-2 and GNOME Hints support.
- Full internationalization support, including multi-byte characters and bidirectional text.
- Xft2 font support with anti-aliasing, drop shadows of any size, at any offset and at any direction, text rotation.
- Title bars can be disabled, or rendered on any window edge.

- Animated Window Shading in all directions.
- Full PNG Support, including alpha blending.
- Perl library for extending FVWM using Perl, scripting and pre-processing of configuration files.
- Can be extended via scripting. Preprocessing allows dynamic configurations.
- Toolkit to build dialogs menus and applications at runtime.
- Configurable desktop panels.
- Mouse Gestures allow drawing shapes with the mouse, and binding them to commands.
- Dynamic menus; utilities to browse the filesystem; fetch headlines from the internet from menus included.
- Session management support.
- Xinerama extension support to use more than one monitor.
- Dynamically extensible using modules.

You can get more information at http://www.fvwm.org/.

The configuration files are at the directory **/usr/share/fvwm**.

Example 9: Refer to the file—configuration example_fvwm.

3-4-20 lbreakout

This is a breakout-style game for Linux.

You can get more information at http://lgames.sourceforge.net/.

3-4-21 ltris

This is a Linux game of tetris clone.

You can get more information at http://lgames.sourceforge.net/.

3-4-22 mp3play

This is an application for playing MP3 data on embedded systems. Mp3play is designed to be able to play mp3 data files within the local filesystem (which can be a network filesystem), or fetch files to play via http over a network. For the http case the address can specify a streaming MP3 server.

Simple usage is :
mp3play file.mp3

3-4-23 rdesktop

rdesktop is a client for Microsoft Windows NT Terminal Server, Windows 2000 Terminal Services, Windows 2003 Terminal Services/Remote Desktop, Windows XP Remote Desktop , and possible other Terminal Services products. Rdesktop currently implements the RDP version 4 and 5 protocol.

You can get more information at http://www.rdesktop.org/.

Chapter 4 SDK (Software Development Kits)

The JetBox SDK has all the required software and utilities for you to develop your own applications. It includes a set of cross-compile toolchain , all the libraries ,header files and some sample codes.

4-1 Installing SDK

JetBox SDK must be installed on your host computer running Linux with glibc 2.3.x. We have confirmed that **Fedore Core 6** Linux distribution can be used to install the tool chain.
Using the command **tar** to uncompress the SDK archive jetbox_sdk-0.82.tgz at the directory where you want to complete the installation.

```
cd /usr/src
```

```
tar xfz jetbox_sdk-0.82.tgz
```

4-2 Writing your application

Writing applications for the JetBox is the same as writing applications for a Linux PC. The better way to develop your own application is to write and test your own applications on a Linux PC first and then port the application from the Linux PC to the JetBox.

Following is the steps to develop an application on a Linux PC and port to the JetBox.

Developing on a Linux PC

1. Create a directory under **ap_src_dir/** and put the source code into this directory.
2. Create a makefile (a controlling file) in **ap_src_dir/** for this application. The makefile extension must be "**.gzmk**".
3. Use the command **make** on the top directory of SDK. The application will be built and the executable binary will be generated in the directory **root_dir/**.

Porting to the JetBox

4. Put the executable binary into the JetBox booting CF card.

In the JetBox SDK, there are 4 example codes for your reference

gzip-1.3.9/ is the source code of GNU gzip version 1.3.9
wget-1.10.2/ is the source code of GNU wget version 1.10.2
gzmx_io-0.9.3/ is a sample code for accessing the GPIO of Jetbox
rstest-1.0.0/ is a simple testing application for RS-232 port

gzip.gzmk, **wget.gzmk**, **gzmx_io.gzmk**, and **gzmx_uart.gzmk** are the makefiles for these applications.

Example 10: Refer to the file—makefile example_gzmx_io.

4-3 Sample programs

There are some sample programs in the JetBox SDK as example codes for the programming API of JetBox. You can take these example codes as a reference to develop your own programs.

4-3-1 DIO access

“gzmx_io.c” is the example code for DIO access.

Example 11: Refer to the file—sample code_gzmx_io.c.

Following is the content of the sample code_gzmx_io.c

```
/***** begin of gzmx_io.c *****/

#include <termios.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/signal.h>
#include <sys/types.h>
#include <sys/ioctl.h>
#include <sys/stat.h>



#define GZMX_IOFILE"/dev/gzmx_io"


typedef unsigned char   u8;
typedef unsigned short  u16;
typedef unsigned long   u32;



#define GZMX_IOC_MAGIC  0xBA
#define DI_SECTION 0
#define DO_SECTION 1
#define DIP_SECTION2

struct gzmx_iocmd {
    u8      cmd;
```

```
    u8      sec_idx;    /* 0        -> DI    */
                /* 1        -> DO    */
                /* 2        -> DIP   */
    u16  dio_no;
    u32     timeout;    /* reserved         */
    u8      data[16];   /*              */
};

#define SET_DO      0x01    /* set DO pin         */
#define CLR_DO      0x02    /* clear DO pin       */
#define TOGGLE_DO   0x03    /* toggle DO pin*/
#define GET_DATA    0x04    /* get data       */
#define SET_DATA    0x05    /* set data       */
#define ENABLE_DO   0x06 /* enable DO output      */
#define DISABLE_DO  0x07 /* disable DO output     */


#define GZMX_IOC_SET_DO _IOW(GZMX_IOC_MAGIC,SET_DO,struct gzmx_iocmd)         /*
Single entity I/O access command      */
#define GZMX_IOC_CLR_DO _IOW(GZMX_IOC_MAGIC,CLR_DO,struct gzmx_iocmd)         /*
Single entity I/O access command      */
#define GZMX_IOC_TOGGLE_DO    _IOW(GZMX_IOC_MAGIC,TOGGLE_DO,struct gzmx_iocmd)/*
Single entity I/O access command      */
#define GZMX_IOC_GET_DATA     _IOWR(GZMX_IOC_MAGIC,GET_DATA,struct gzmx_iocmd)/*
Single entity I/O access command      */
#define GZMX_IOC_SET_DATA     _IOWR(GZMX_IOC_MAGIC,SET_DATA,struct gzmx_iocmd)/*
Single entity I/O access command      */
#define GZMX_IOC_ENABLE_DO    _IO(GZMX_IOC_MAGIC,ENABLE_DO)                   /*
enable DO output      */
#define GZMX_IOC_DISABLE_DO   _IO(GZMX_IOC_MAGIC,DISABLE_DO)                  /*
enable DO output      */


int main( int argc, char *argv[] ) {
    int i,fd;

    struct gzmx_iocmd iocmd;
```

```
	printf ("Test program for GZ Media-X gpio 3¥n");

	fd = open( GZMX_IOFILE, O_RDWR | O_NOCTTY | O_NONBLOCK);

	if (fd <0) {perror(GZMX_IOFILE); exit(-1); }

	printf ("gzmx_io file opened ¥n");


//printf("enable do %d¥n",GZMX_IOC_ENABLE_DO);
	ioctl(fd,GZMX_IOC_ENABLE_DO);



	printf ("¥n get data from di section¥n");

	iocmd.sec_idx = DI_SECTION;
	ioctl (fd,GZMX_IOC_GET_DATA,&iocmd);

	printf (" DI a%08x: %02x %02x¥n",GZMX_IOC_GET_DATA,iocmd.data[0],iocmd.data[1]);


	printf ("¥n get data from dip section¥n");

	iocmd.sec_idx = DIP_SECTION;
	ioctl (fd,GZMX_IOC_GET_DATA,&iocmd);

	printf (" DIP %08x : %02x %02x %02x
%02x¥n",GZMX_IOC_GET_DATA,iocmd.data[0],iocmd.data[1],iocmd.data[2],iocmd.data[3]);


	printf ("¥n get current data of do section¥n");

	iocmd.sec_idx = DO_SECTION;
	ioctl (fd,GZMX_IOC_GET_DATA,&iocmd);

	printf (" DO %08x: %02x %02x ¥n",GZMX_IOC_GET_DATA,iocmd.data[0],iocmd.data[1]);
```

```
    printf ("¥n test on do section¥n");

    while(1) {
        for (i = 0; i < 16 ; i++) {
            iocmd.sec_idx = DO_SECTION;
            iocmd.dio_no= i;
            ioctl (fd,GZMX_IOC_CLR_DO,&iocmd);
            printf ("set do %d high¥n",i);
            sleep(1);
            ioctl (fd,GZMX_IOC_SET_DO,&iocmd);
//          sleep(1);
        }
    }

    printf ("test complete ¥n");

    exit(0);
}

/***** End of gzmx_io.c *****/
```

4-3-2 RS-232/422/485 control

"gzmx_uart.c" is an example code to demo the controlling of serial modes (RS-232/RS-422/RS-485).

Example 12: Refer to the file—sample code_gzmx.uart.c

Following is the content of sample code_gzmx.uart.c.

```
/***** Begin of gzmx_uart.c *****/
#include <termios.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
```

```
#include <sys/signal.h>
#include <sys/types.h>
#include <sys/ioctl.h>
#include <sys/stat.h>



#define GZMX_UARTFILE"/dev/gzmx_io"


typedef unsigned char      u8;
typedef unsigned short     u16;
typedef unsigned long      u32;



#define GZMX_IOC_UART_MAGIC  0xBC

#define GZMX_IOC_UART_SET  _IOW(GZMX_IOC_UART_MAGIC,0,unsigned short)
#define GZMX_IOC_UART_GET  _IOR(GZMX_IOC_UART_MAGIC,0,unsigned short)


int main( int argc, char *argv[] ) {
      int i,fd;

      printf ("Test program for GZ Media-X gpio 3¥n");

      fd = open( GZMX_UARTFILE, O_RDWR | O_NOCTTY | O_NONBLOCK);

      if (fd <0) {perror(GZMX_UARTFILE); exit(-1); }

      printf ("gzmx_io file opened ¥n");

      printf ("¥n get data from di section¥n");

      iocmd.sec_idx = DI_SECTION;
      ioctl (fd,GZMX_IOC_GET_DATA,&iocmd);
```

```
	printf (" DI a%08x: %02x %02x¥n",GZMX_IOC_GET_DATA,iocmd.data[0],iocmd.data[1]);


	printf ("¥n get data from dip section¥n");

	iocmd.sec_idx = DIP_SECTION;
	ioctl (fd,GZMX_IOC_GET_DATA,&iocmd);

	printf (" DIP %08x : %02x %02x %02x
%02x¥n",GZMX_IOC_GET_DATA,iocmd.data[0],iocmd.data[1],iocmd.data[2],iocmd.data[3]);


	printf ("¥n get current data of do section¥n");

	iocmd.sec_idx = DO_SECTION;
	ioctl (fd,GZMX_IOC_GET_DATA,&iocmd);

	printf (" DO %08x: %02x %02x ¥n",GZMX_IOC_GET_DATA,iocmd.data[0],iocmd.data[1]);


	printf ("¥n test on do section¥n");

	for (i = 0; i < 16 ; i++) {
		iocmd.sec_idx = DO_SECTION;
		iocmd.dio_no= i;
		ioctl (fd,GZMX_IOC_CLR_DO,&iocmd);
		printf ("set do %d high¥n",i);
		sleep(2);
		ioctl (fd,GZMX_IOC_SET_DO,&iocmd);
		sleep(2);
	}


	printf ("test complete ¥n");

	exit(0);
}
```

```
/***** End of gzmx_uart.c *****/
```

Chapter 5 Appendix

5-1 Notice, Chart & Example Index

Notice

Notice 1: If the image of the CF card is incorrect or there is no CF card in the JetBox, the splash screen shows a warning message “check CF card??” 10

Chart

Chart 1: The JetBox 8210 Linux SW specification 6
Chart 2: The JetBox 8210 Linux SW service list..... 8
Chart 3: The default RS-232 setting of the JetBox 8210 LInux..... 9
Chart 4: The default IP address setting of the JetBox 8210 Linux..... 10

Example

Example 1: Refer to the file—configuration example_snmpd. 21
Example 2: Refer to the file—configuration example_openvpn. 22
Example 3: Refer to the file—configuration example_openswan(ipsec). . 23
Example 4: Refer to the file—configuration example_pppoe. 24
Example 5: Refer to the file—configuration example_proftpd. 25
Example 6: Refer to the file—configuration example_samba. 25
Example 7: Refer to the file—configuration example_php. 27
Example 8: Refer to the file—configuration example_httpd. 27
Example 9: Refer to the file—configuration example_fvwm. 30
Example 10: Refer to the file—makefile example_gzmx_io. 32
Example 11: Refer to the file—sample code_gzmx_io.c. 33
Example 12: Refer to the file—sample code_gzmx.uart.c 36

5-2 Customer Service

Korenix Technologies Co., Ltd.
9F, No. 100-1, Ming-Chuan Rd., Shing Tien City, Taipei, Taiwan

Tel:+886-2-82193000 Fax:+886-2-82193300

Business service: sales@korenix.com

Customer service: koreCARE@korenix.com